Bölüm 3: Örnek Uygulamalar

JAVA ile Nesne Yönelimli Programlama

Verilen Bir Sayı Tek Mi Çift Mi Bulma

- Tek Sayılar: 1, 3, 5, 7, 9... gibi sayılar, 2 ile bölündüğünde kalan 1'dir.
- Çift Sayılar: 2, 4, 6, 8, 10... gibi sayılar, 2 ile tam bölünebilir.
- Tek mi Çift mi Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 2 ile bölündüğünde kalan 0 ise, sayı çifttir.
  - Kalan 1 ise, sayı tektir.
  - Koşul: sayi % 2 == 0 ise çift, değilse tek.

Verilen Bir Sayı Tek Mi Çift Mi Bulma

```
// Modulus (%) operatörü ile sayının 2'ye bölümünden kalan kontrol edilir.
// Eğer kalan 1 ise sayı tektir.
if(sayi % 2 == 1) {
  // sayı tektir, bu mesaj ekrana yazdırılır.
  System.out.println("Girilen sayı tektir.");
}
// Eğer sayı teklik şartını sağlamazsa, sayı çifttir.
else {
  // sayı çifttir, bu mesaj ekrana yazdırılır.
  System.out.println("Girilen sayı çifttir.");
}
```

Basamak Sayısı Bulma

- Bir sayının basamak sayısı, sayının içindeki toplam rakam sayısıdır.
- Örneğin:
  - 123 sayısı 3 basamaklıdır (1, 2, 3).
  - 45 sayısı 2 basamaklıdır (4, 5).
- Basamak Sayısını Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 0'dan büyük olduğu sürece, sayının her bir basamağı silinir.
  - Her bölme işleminde basamak sayısı bir artırılır.
  - Koşul: while (sayi > 0).

Basamak Sayısı Bulma

```
// Basamak sayısını tutmak için bir değişken tanımlanır ve başlangıç değeri 0 olarak belirlenir.
int basamakSayisi = 0;

// 'while' döngüsü, sayının basamaklarını saymak için kullanılır.
// Döngü, sayı 0'dan büyük olduğu sürece devam eder.
while(sayi > 0) {

  // Sayının son basamağı "sayı / 10" ile silinir (örneğin 1234 -> 123).
  sayi = sayi / 10;

  // Her işlemde bir basamak silindiğinden 'basamakSayisi' bir artırılır.
  basamakSayisi++;
}
```

Verilen Bir Sayının Rakamlar Toplamı

- Bir sayının rakamları toplandığında elde edilen sonuç.
- Örneğin:
  - 123 sayısının rakamları toplamı: 1 + 2 + 3 = 6
  - 456 sayısının rakamları toplamı: 4 + 5 + 6 = 15
- Rakamlar Toplamını Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 0'dan büyük olduğu sürece,
    - sayının son basamağını alarak toplamına eklenir.
  - Sayının son basamağını silmek için sayı 10'a bölünür.
  - Koşul: while (sayi > 0).

Verilen Bir Sayının Rakamlar Toplamı

```
int rakamlarToplami = 0;
// Sayının basamaklarını tek tek almak için bir döngü başlatılır.
// Döngü, 'sayi' 0'dan büyük olduğu sürece devam eder.
while(sayi > 0) {
  // Sayının son basamağını almak için 10'a göre mod (kalan) yapılır.
  int rakam = sayi % 10;
  // Alınan basamak değeri 'rakamlarToplami' değişkenine eklenir.
  rakamlarToplami += rakam;
  // Sayının son basamağını silmek için sayı 10'a bölünür.
  // Bu işlemle sayının bir basamağı eksilir.
  sayi = sayi / 10;
}
```

Verilen Bir Sayının Üssünü Alma

- Bir sayının kendisiyle belirli sayıda çarpılması işlemine denir.
- Örneğin:
  - 2^3 = 2 × 2 × 2 = 8
  - 5^2 = 5 × 5 = 25
- Üstü Alma Algoritması
  - Kullanıcıdan bir taban ve üs alınır.
  - Sonuç değişkeni başlangıçta 1 olarak ayarlanır.
  - Üs kadar döngü oluşturulur.
  - Her döngüde sonuç değişkeni ile taban çarpılır.

Verilen Bir Sayının Üssünü Alma

```
int sonuc = 1; // Sonucu hesaplamak için başlangıç değeri 1 olan bir
'sonuc' değişkeni tanımlanır.

for(int i = 0; i < us; i++) { // Döngü, üs sayısı kadar tekrar eder.
  sonuc = sonuc * taban; // Her adımda 'sonuc', 'taban' ile çarpılır.
}
```

Faktöriyel Hesaplama

- Pozitif bir tam sayının, kendisinden küçük pozitif tam sayıların çarpımıdır.
- Matematiksel notasyon: n! (n faktöriyel)
- Örneğin:
  - 5! = 5 × 4 × 3 × 2 × 1 = 120
  - 3! = 3 × 2 × 1 = 6
- Faktöriyel Hesaplama Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Faktöriyel başlangıçta 1 olarak ayarlanır.
  - Sayı 0'dan büyük olduğu sürece döngü çalışır,
  - Her döngüde faktöriyel değişkeni ile sayı çarpılır, ardından sayı azaltılır.

Faktöriyel Hesaplama

```
// Döngü, sayı 0'dan büyük olduğu sürece devam eder.
// Faktöriyel, 1'den başlayarak sayıya kadar olan tüm sayıların çarpımına
eşittir.
while(sayi > 0) {

  // 'faktoriyel' değeri, her adımda o anki 'sayi' değeri ile çarpılır.
  faktoriyel *= sayi;

  // Sayıyı 1 azaltarak bir sonraki sayıya geçer.
  sayi--;
}
```

Asal Sayı Bulma

- 1 ve kendisi dışında pozitif tam böleni olmayan sayılara asal sayı denir.
- Örnekler: 2, 3, 5, 7, 11, 13...
- Not: 1 asal sayı değildir, 2 ise tek asal sayıdır.
- Asal Sayı Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - 2'den başlayarak, verilen sayıya kadar olan her sayıya tam bölünüp bölünmediği kontrol edilir.
  - Eğer sayı herhangi bir bölenle tam bölünmüyorsa, asal sayıdır.

Asal Sayı Bulma

```
// Başlangıçta sayının asal olduğu varsayılır.
boolean asal = true;
// Döngü 2'den başlayarak girilen sayıya kadar devam eder.
for(int i = 2; i < sayi; i++) {
  // Eğer sayı, 2 ile 'sayi' arasındaki herhangi bir sayıya tam bölünürse, asal
değildir.
  if(sayi % i == 0) {
    // Eğer sayının bir böleni bulunursa, ekrana "sayı asal değil" yazdırılır.
    System.out.println("Sayı asal değil.");
    // Sayının asal olmadığını belirtmek için 'asal' değişkeni false yapılır.
    asal = false;
    // İlk bölenden sonra kontrol etmeye gerek olmadığı için döngüden çıkılır.
    break;
  }
}
```

Rastgele Sayı Üretme

- Belirli bir aralıkta rastgele olarak seçilen sayılardır.
- Rastgele sayıların kullanımı, oyunlar, simülasyonlar, şifreleme ve istatistik gibi birçok alanda yaygındır.
- Rastgele Sayı Üretme Yöntemleri
  - *Gerçek Rastgele Sayılar*: Fiziksel süreçlere (örneğin, radyoaktif bozunma) dayalıdır.
  - *Pseudo Rastgele Sayılar*: Matematiksel algoritmalarla üretilir ve belirli bir başlangıç noktasına (seed - tohum) dayanır.
  - Bilgisayar programlamada genellikle pseudo rastgele sayı üreteçleri kullanılır.

Rastgele Sayı Üretme

```
// Random sınıfından bir nesne oluşturulur.
Random rastgele = new Random();

// 'nextInt(100)' ifadesi 0-99 arasında bir sayı üretir,
// bu yüzden sonuca 1 eklenir ve 1-100 aralığında bir sayı elde edilir.
int sayi = rastgele.nextInt(100) + 1;
```

Öğrenci Numarasını Çözme

- Öğrenci numarası, her öğrenciyi tanımlamak için kullanılan tekil kimliktir.
- Genellikle bir dizi rakamdan oluşur ve öğrencinin kayıt bilgilerini içerir.
- Öğrenci numarası, öğrencinin kayıt bilgilerini hızlı bir şekilde çözümler.
- Öğrenci Numarasının Yapısı
  - İlk iki hane: Kayıt yılı (örneğin, 24, 23 vb.)
  - Beşinci hane: Bölüm kodu (0-9 arası, belirli bölümleri temsil eder)
  - Son üç hane: Kayıt sırası (her bölümdeki kayıt sırası numarası)

Öğrenci Numarasını Çözme

```
// İlk iki hane kayıt yılını belirler. Bu iki hane 10000000'e bölünerek elde edilir. 2000 yılına eklenerek öğrenci numarasındaki yıl bilgisi çözülür.
int kayityili = (numara / 10000000) + 2000;
// Son üç hane, kayıt sırasını gösterir. Mod 1000 işlemi ile bu üç hane alınır.
int kayitsirasi = numara % 1000;
// 5. hane öğrencinin bölümünü gösterir. 100000'e bölünür ve ilgili hane mod 10 işlemi ile elde edilir.
int bolumu = (numara / 100000) % 10;
// Numarasına göre öğrencinin hangi bölümde olduğu belirlenir.
if(bolumu == 7) {
    System.out.println("Bölüm: Elektrik Elektronik");
} else if (bolumu == 6) {
    System.out.println("Bölüm: Bilgisayar");
} else {
    System.out.println("Bilinmeyen bölüm");
}
```

Fibonacci Serisi Yazdırma

- Fibonacci serisi, her sayının kendisinden önceki iki sayının toplamı olduğu bir sayı dizisidir.
- Seri genellikle 1 ve 1 ile başlar: 1, 1, 2, 3, 5, 8, 13, 21, ...
- Matematiksel olarak tanımı:
  - F(0) = 1
  - F(1) = 1
  - F(n) = F(n-1) + F(n-2) (n ≥ 2)
- Doğada pek çok yerde karşımıza çıkar (bitki yapraklarının düzeni).
- Altın oran ile bağlantılıdır; dizinin ilerleyen sayıların oranı 1.618'e yaklaşır.

Fibonacci Serisi Yazdırma

```
int sonuc = 0;
// Döngü, ilk iki eleman belli olduğundan, 2. indisten başlar
for(int i = 2; i < sayi; i++) {
  // 'sonuc', bir önceki iki Fibonacci sayısının toplamı.
  sonuc = sayi1 + sayi2;
  // 'sayi1' bir sonraki adım için 'sayi2' olur.
  sayi1 = sayi2;
  // 'sayi2' yeni hesaplanan Fibonacci sayısı olan 'sonuc' olur.
  sayi2 = sonuc;
  // Altın oran, ardışık iki Fibonacci sayısı arasındaki oran.
  System.out.println("Altın oran = " + (double)((double)sayi2 / sayi1));
}
```

SON